

Study of Computer Malware and Its Taxonomy

Bishir Suleiman, Rashid Husain

Abstract— There are widespread availability and use of information technology devices capable of accessing the Internet from any remote location. While communicating via the Internet, each of these devices face a major challenge - malicious codes that alter their actual behavior, function, etc. Malicious codes result in heavy damages to computer owners, users, and the society at large. This study describes the general architecture of computer malware. It focuses on the classification of computer malware using three (3) suitable criteria. It proposes the likely properties of future computer malware going by the trend in their development. Our future work would focus on developing a model for worm-attack and simulating it against campus network. It choose computer worms because of the enormous devastating effects they cause as commonly experienced on computers connected via network.

Index Terms— Malicious codes, Architecture, Malware, Computer malware, Worm, Worm-attack, Simulating.

I. INTRODUCTION

Malicious codes, commonly called malware are any code fragments that are either added, changed or removed from a software system in order to intentionally cause havoc or subvert the intended function of the system. Though the problem of malicious codes has a long history, a number of recent widely publicized attacks and certain economics trend suggest that malicious codes are rapidly becoming a critical problem for industry, government, and individuals (Abuzaid et al., 2013).

We more often experience a sudden change in the type of installed default browser in use initially, slow computer speed or performance, computer system freezes and presents blue screens of death, continuous self-rebooting of computer systems, erasure of entire disk or drive, erratic screen behavior, browser's homepage changed itself, modified operating system software, etc. (Milind & Patil, 2013). All these points to some of the symptoms of computer malware. Some computer malware depend on network media for their widespread transmission while others do not. Generally, computer malware first infects a vulnerable system, and then spreads to one or more systems; it then becomes active in the host computer depending on the nature of its coding content and finally subverts the host system in the way it is intended. This paper is aimed at describing the general architecture of computer malware. We further classify and discuss each category of computer malware using three (3) suitable criteria namely: transmission media, nature of damage and intelligence. Also, we highlight the major features expected of future computer malware in order to avoid being detected and appropriately handled intrusion detection systems. In this

paper, the terms 'malware' and 'malicious codes' are used interchangeably.

II. GENERAL ARCHITECTURE OF COMPUTER MALWARE

Generally, Computer malware affect computing devices through the following four (4) stages:

1. Infection mechanism used
 2. Propagation/Spreading mechanism employed
 3. Activation mechanism used
- Nature of Attack

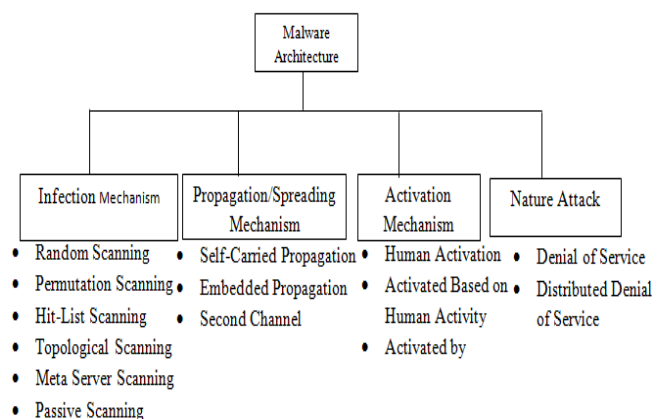


Figure 1: Malware Architecture

1. Infection Mechanism used: Computer malware finds new host to infect using a number of approaches some of which include:

- **Random Scanning:** Here, the malware randomly selects the vulnerable host and probes it. The malware continues to generate new vulnerable hosts at random (Ashfa et al., 2008).
- **Permutation Scanning:** Permutation scanning is an improvement over random scanning such that the malware now avoids probing the same address multiple times and coordinates the scanning of many infected hosts (Stuart & Nicholas, 2002).
- **Localized Scanning:** This is a network-dependent infection technique employed by computer malwares for the sole aim of infecting hosts that posses local network addresses (Vogt, 2003).
- **Hit-list scanning:** Malwares can also use a pre-generated list of potential hosts, known as a hit-list, to speed the rate of initial infections. The hit-list usually contains a list of addresses which are likely running vulnerable services. This hit-list can then be split up and distributed to newly infected hosts (Moheeb et al., 2006).
- **Topological scanning:** Malwares which employ topological scanning gather potential hosts from the local machine. This includes the email addresses in a user's contact list, URLs in the user's browsing history, etc.

Bishir Suleiman, Research Scholar, Department of Mathematics and Computer Science, Faculty of Natural and Applied Sciences, Umaru Musa Yar'adua University, Katsina, Katsina-State, Nigeria

Rashid Husain, Lecturer, Department of Mathematics and Computer Science, Faculty of Natural and Applied Sciences, Umaru Musa Yar'adua University, Katsina, Katsina-State, Nigeria

- Metaserver scanning: Malwares can query a Metaserver to find potential hosts, such as the services provided by Google, Gamespy or Netcraft, or by querying a peer-to-peer network or an instant messaging server for vulnerable peers.
- Passive scanning: A different, network-based and less common approach is for the malware to passively wait for incoming or outgoing connections and extract information from these connections to determine new hosts.

2. Propagation/Spreading mechanism employed: Basically, there are three (3) main malware propagation mechanisms. They are:

- Self-carried propagation: Self-carried malware spread or transmit itself during initial communication with the vulnerable host.
- Embedded propagation: Here, malwares transmit themselves within normal communication channels either by appending themselves to normal messages, or replacing normal messages (Marco et al., 2006).
- Secondary channel propagation: Here, malware rely on a secondary channel for its propagation. Such malwares infect vulnerable hosts in two (2) stages, i.e. it sends driver program of the malware which will subsequently download and run the rest of the malware.

3. Activation mechanisms used: There are four methods by which computer malwares are triggered to display their behaviors in infected hosts. They are:

- Human activation: Certain malwares depend on human beings (as users of IT devices) to manually execute their programs. Malwares which are activated when a user clicks on an email or which copy infected files onto a shared folder fall into this category.
- Activated based on human activity: Here, computer malware are activated by a user's actions which wouldn't normally be expected to execute a worm, such as via a user's login scripts, or when a CD or memory card is inserted into the computer.
- Activated by scheduled processes: Malware are activated by a legitimate automated process which hasn't been properly secured, such as a legitimate program which automatically updates it-self from an infected web server.
- Self activation: These are the most worrisome type of computer malware that begin execution immediately after being transmitted to the vulnerable host. These malwares generally take advantage of one or more vulnerabilities in a running application.

4. Nature of Attack: Computer malware display a variety of behaviors after it had attacked a vulnerable host. Some malware display their effect immediately it is activated while many others stay resident in their host for a long time before causing severe damages to hosts. The nature of attack caused by malwares includes controlling, data theft, modifying or encrypting files on infected hosts. Malwares may also cause physical damage such as re-flashing a host's BIOS, etc (Weaver et al., 2003).

III. CATEGORIES OF NATURE OF ATTACK

- Semi hybrid: Is a form of attack that combined the behaviors of short-term and long-term (Georgios and Herbert 2007)
- Hybrid:

IV. TAXONOMY OF COMPUTER MALWARE

We classify computer malware according to the following criteria:

1. Malware transmission media.
2. Nature of damage caused by malware.
3. Malware intelligence.

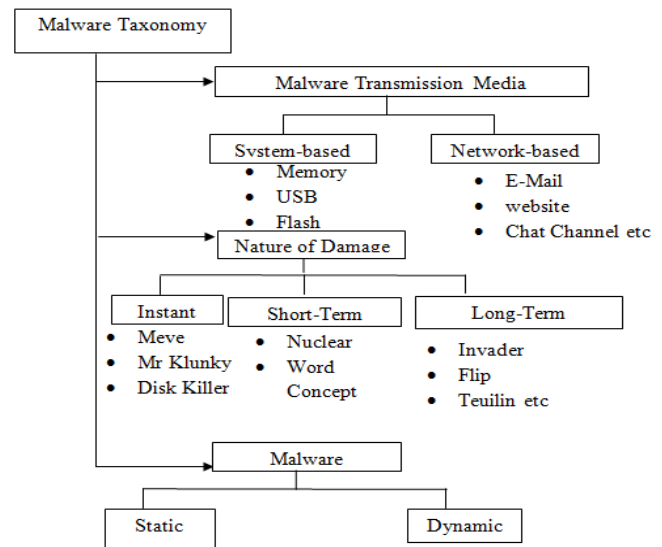


Fig.2: Malware Taxonomy

1. Malware transmission media: Many computer malwares depend on particular medium for them to permeate into I.T. devices. Based on this criterion, we classify computer malware as either system-based or network-based. System-based malware become manifest when I.T. devices are in contact with other infected devices most probably in the process of sharing required documents. The media used by system-based malware include: memory sticks, smart phones, flash drives, etc. Alternatively, network-based malware rely on the underlying network infrastructure to spread across interconnected systems thereby causing large scale havoc when compared to the disastrous effect caused by system-based malware. In comparison, network-based malware are self-replicating whereas system-based malware require user action or activity in order to initiate the usual process of infection. Examples of system-based malware include: Randex Virus, cascade, Disk Killer, Stone Virus, etc. Examples of network-based malware include: Melissa, ILOVE YOU, Love Bug, Lovgate F, Trile C, A2KM Nitrogen, 8sec!Trojan, Conficker Worm, CR Clean Worm, etc (Essam et al., 2008).

2. Nature of damage caused by malware: Various malware are coded in such a way that they specify the time they begin to really affect infected systems. Some malware portray their effect as soon as the system under consideration becomes infected while others stay within the infected system for a while before showing its true nature. According to this, we classify the nature of damage caused by malware as instant, short-term or long-term. A malware that becomes active as soon as it infects a host computer system is said to be instant. Instant malware usually infect, propagate and becomes active simultaneously hence, they present an architecture that lacks clear-cut boundaries across phases. Short-term malware take a few days resident in its host before becoming active while long-term malware stays in its host for at least 30 days before it begins to display signs of controlling or taking charge of its

host. The delay by short-term and long-term malwares to display the nature of their effect within a host may be as a result of the code that specifies the time to be active. For malware that depend on particular type of human activity, this delay might be experienced if no such activity has taken place for a longer period of time. Instant malwares are: Meve, MrKlunky, Disk Killer, Stone-virus, etc, short-term malwares are: Nuclear, word concept, etc and long-term malwares are: Invader, Flip, Tequila, etc.

3. Malware intelligence: Malware can reprogram itself. It use code obfuscation techniques to challenge deeper static analysis and can also beat dynamic analyzers by altering its behavior, it does this by transmitting its own code into a temporary representation, edit the temporary representation of itself, and then write itself back to normal code again. This procedure is done with the malware itself, and also the malware engine itself undergoes changes malware use several transformation techniques or measures including:

- Instruction reordering
- Data reordering
- Inlining
- Outlining
- Register renaming
- Code permutation
- Code expansion
- Code shrinking
- Subroutine interleaving
- And garbage code insertion.

The altered code is then recompiled to create a malware executable that looks fundamentally different from the original (Szor, 2005).

Some malware have fixed number of features they display after infecting a host computer while others change from one nature of effect to another in order to avoid being detected by available intrusion detection systems. According to malware intelligence, we classify malware as static or dynamic. Static malware has only one mode or nature of attack that can easily be handled using appropriate intrusion detection systems. On the other hand, dynamic malware have a way of changing its nature from one form to another, such that intrusion detection systems find it difficult to identify and handle it (i.e. disable, delete, etc) appropriately. Some research works claim that dynamic malware only changes its shape but its architecture remains the same throughout its life span. Static Malwares are Metamorphic virus and dynamic malwares are Polymorphic and Oglimorphic.

V. FUTURE COMPUTER MALWARE

Future computer malware mainly attempts to incorporate sophisticated intelligence such that malware effects are felt but cannot be traced by intrusion detection systems. This would mean that a future malware may have as much architecture as its number of attack-types so that its multiple natures would enable it cause havoc to many vulnerable system in unit time instance.

Future malware would likely have a unique key for each of its architectures and an overall identifier key that signify its entire characteristics. While changing from one architecture, shape, etc to another, it increases the difficulty of being detected and appropriately handled.

Another dimension to the likely future of malware is in the area of P2P networks. Future malware could simply take advantage of P2P overlay to rapidly propagate across different interconnected systems thereby causing severe damage.

Intrusion detection systems on one part and entire system security on the other hand must develop to counter the ills of future malicious codes.

VI. CONCLUSION

In this paper, it has described the phases that collectively define the general architecture for computer malware. Ideally, a computer malware attempts to infect vulnerable systems. When it is successful, the malware begins to spread to other vulnerable systems. Depending on the time it is coded to take control of its host computer, it becomes active and displays all forms of its unwanted behavior. We also looked at the nature of attack by computer malware. Further, we classify and discuss each category of computer malware using three (3) suitable criteria namely: transmission media, nature of damage and intelligence. Also, we highlight the major features expected of future computer malware in order to avoid being detected and appropriately handled using intrusion detection systems. Future intrusion detection systems must also take the direction of intelligent-based systems in order to counter the likely threat that next generation malware would cause.

REFERENCES

- [1] Abuzaid A. M., Saudi M. M., Taib B. M., & Abdullah H. (2013). "An efficient Trojan Horse Classification (ETC)", *International Journal of Computer Science*, Vol. 10 (2), pp: 1694-1704.
- [2] Ashfa A., B., Robert M. J., Mumtaz A., Muhammad Q. A., Ali S. and Syed Ali Khayam, (2008). "A Comparative Evaluation of Anomaly Detectors Under Portscan Attacks In Recent Advances in Intrusion Detection", *11th International Symposium, RAID 2008, Cambridge, Massachusetts, U.S.A.*, pp: 351-371.
- [3] Essam A., Iqbal H. J., & Belal Z., (2008). "Computer Strategies and Detection Method", *International Journal Open Problem Computer, Mathematics*, Vol. 1(2).
- [4] Georgios P. and Herbert B., (2007). "SweetBait: Zero-hour worm detection and containment using low-and high-interaction honeypots", *Computer Networks: The International Journal of Computer and Telecommunications Networking*, Vol. 51(5), pp: 1256-1274.
- [5] Marco B., Blaine N., Russell S., An-thony D. J., and J. D. Tygar, (2006). "Can machine learning be secure?", *In ASIACCS '06: Proceedings of the 2006 ACM Symposium on Information, computer and communications security*, New York, NY, USA. ACM Press, pp: 16-25.
- [6] Milind J.J., & Patil B. V., (2013). "Computer Virus their Problems and Major Attack in Real life", *International Journal of P2P network trends and Technology (IJPTT)*, Vol. 4.
- [7] Moheeb A. R., Fabian M., & Andreas T., (2006). "Fast and Evasion Attacks. Highlighting the Challenges Ahead", *In recent Advances Intrusion Detection 9th International Symposium, Hamburg-Germany*, pp: 206-225.
- [8] Weaver V., Paxson V., Staniford S., Cunningham J., (2003). "A Taxonomy of Computer Worms", *In First ACM Workshop on Rapid Malcode*.
- [9] Stuart S., Vern P. & Nicholas W., (2002). "How to Own the Internet in Your Spare time", *In Proceedings of the 11th USENIX Security Symposium*, pp: 149.
- [10] Szor P., (2005). "The Art of Computer Virus Research and Defense", *Addison Wesley*.
- [11] Vogt T., (2003). "Simulating and Optimizing Worm Propagation Algorithms", Available from <http://downloadsssecurityfocus.com/library/worm>.